

作业提交服务器: <ftp://192.168.134.123>

用户名: uploader

密码: sdt7%5252@3

! 注意事项

1. 每题需要提交一个cpp文件，确保cpp文件可以正确编译。cpp文件用题目编号命名，除非有特别说明。
2. 每题需要提交至少一张运行结果截图。将所有题目的截图放在一个pdf文件中。
3. 将所有cpp文件和一个pdf文件打包成zip文件，用自己的学号命名，并上传至ftp指定文件夹。

举例说明：比如学号是1001，本次有3个题目，那么最终提交1个1001.zip文件，其中包含3个cpp文件（分别是1.cpp, 2.cpp, 3.cpp）和1个pdf文件，其中pdf文件中包含至少3个运行结果的截图。第1次作业上传至hm1文件夹。

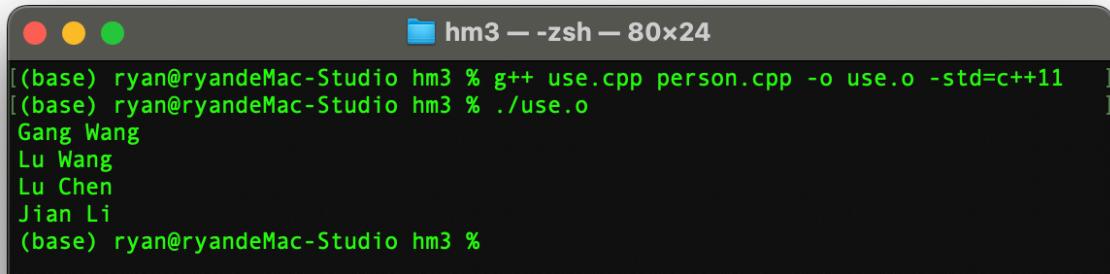
第3次作业

提交时间: 2024年5月5日22:00

1. 编写一个Person类，将名字和姓氏存储为数据成员。类的接口放在person.h文件中，类的实现放在person.cpp文件中。设计一个接受两个参数（名字和姓氏）的构造函数，并增加适当的成员函数使得下面的use.cpp能够正确编译运行。(本题需提交两个代码文件: person.h和person.cpp，以及一个pdf文件)

```
1 #include "person.h"
2 #include <iostream>
3
4 int main()
5 {
6     Person student {"Gang", "Wang"};
7     std::cout << student.get_first_name() << ' ' << student.get_last_name() <<
8     '\n';
9
10    student.set_first_name("Lu");
11    std::cout << student.get_first_name() << ' ' << student.get_last_name() <<
12    '\n';
13
14    student.set_last_name("Chen");
15    std::cout << student.get_first_name() << ' ' << student.get_last_name() <<
16    '\n';
17
18    const Person another_student {"Jian", "Li"};
19    std::cout << another_student.get_first_name() << ' ' <<
20    another_student.get_last_name() << '\n';
21
22    return 0;
23 }
```

程序运行结果截图如下



```
(base) ryan@ryandeMac-Studio hm3 % g++ use.cpp person.cpp -o use.o -std=c++11
(base) ryan@ryandeMac-Studio hm3 % ./use.o
Gang Wang
Lu Wang
Lu Chen
Jian Li
(base) ryan@ryandeMac-Studio hm3 %
```

2. 使用上一题编写的Person类代码，以下代码another_use.cpp能够正确编译吗？如果不能，首先解释原因，然后修改Person类的实现使得如下的another_use.cpp能够正确编译运行。（本题仅需提交一个pdf文件，内容包括：对编译结果进行截图、解释原因、修改后代码截图和编译运行结果截图）

```

1 #include "person.h"
2 #include <iostream>
3
4 int main()
5 {
6     Person students[2];
7     for (auto e : students) {
8         std::cout << e.get_first_name() << ' ' << e.get_last_name() << '\n';
9     }
10
11    students[0].set_first_name("Jian");
12    students[0].set_last_name("Li");
13    students[1].set_first_name("Fei");
14    students[1].set_last_name("Wang");
15
16    for (auto e : students) {
17        std::cout << e.get_first_name() << ' ' << e.get_last_name() << '\n';
18    }
19    return 0;
20 }
```

3. 编写一个Name_pairs类，用来保存名字和年龄对，其中名字是一个string，年龄是一个double。将值对表示成一个名为m_names的vector成员和一个名为m_ages的vector成员。提供一个read_name方法，提示用户输入名字，并以quit结束输入结束标志。提供一个read_age方法，提示用户为每个名字输入一个年龄。提供一个print方法，按name向量的顺序打印名字和年龄对，每行输出一个值对。提供一个sort方法，将name向量按字典序排序，并调整age向量与name向量新顺序匹配。将所有操作实现为成员函数。类的接口放在name_pairs.h文件中，类的实现放在name_pairs.cpp文件中。类的使用放在use_name_pairs.cpp文件中，代码如下所示。（本题与下一题共需提交两个代码文件：name_pairs.h和name_pairs.cpp，以及一个pdf文件）

```

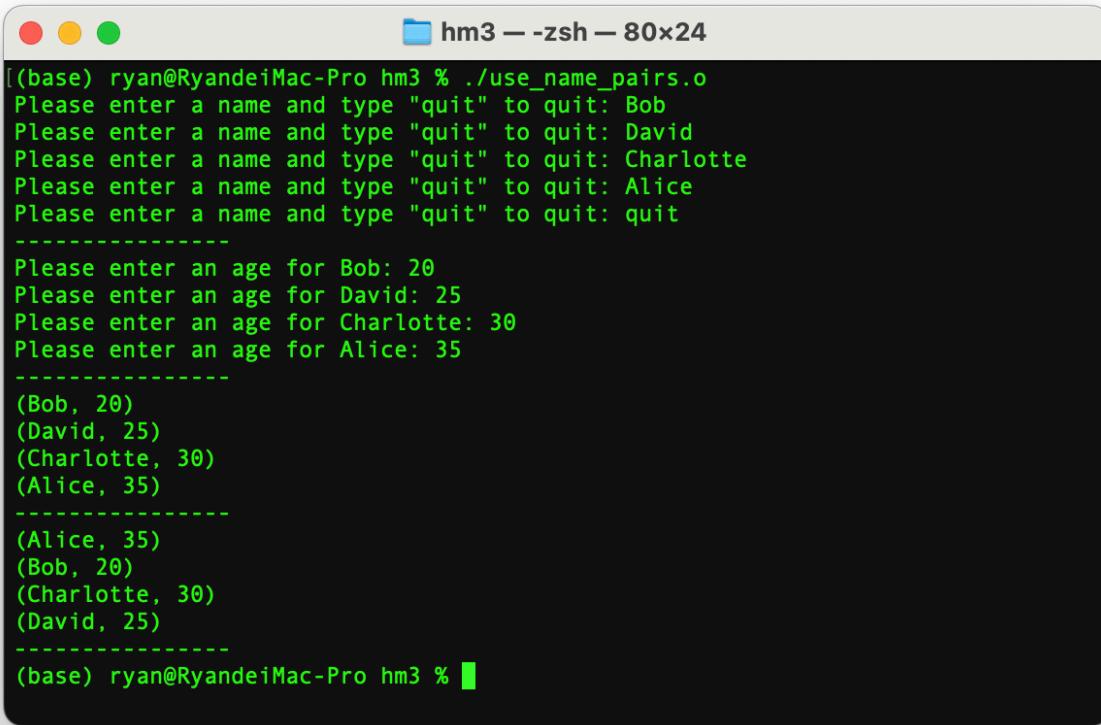
1 #include "name_pairs.h"
2 #include <iostream>
3
4 int main()
5 {
6     Name_pairs np;
7     np.read_names();
8     std::cout << "-----\n";
```

```

9
10    np.read_ages();
11    std::cout << "-----\n";
12
13    np.print();
14    std::cout << "-----\n";
15
16    np.sort();
17    np.print();
18    std::cout << "-----\n";
19    return 0;
20 }

```

程序运行结果截图如下：



```

(base) ryan@RyandeiMac-Pro hm3 % ./use_name_pairs.o
Please enter a name and type "quit" to quit: Bob
Please enter a name and type "quit" to quit: David
Please enter a name and type "quit" to quit: Charlotte
Please enter a name and type "quit" to quit: Alice
Please enter a name and type "quit" to quit: quit
-----
Please enter an age for Bob: 20
Please enter an age for David: 25
Please enter an age for Charlotte: 30
Please enter an age for Alice: 35
-----
(Bob, 20)
(David, 25)
(Charlotte, 30)
(Alice, 35)
-----
(Alice, 35)
(Bob, 20)
(Charlotte, 30)
(David, 25)
-----
(base) ryan@RyandeiMac-Pro hm3 %

```

4. 修改Name_pairs类，将Name_pairs::print()函数替换为运算符<<，并定义==和!=运算符。注意两个Name_pairs对象定义为相等如果它们包含相同的值对。基于修改后的类，编译运行下面的another_use_name_pairs.cpp文件。(本题与上一题共需提交两个代码文件：name_pairs.h和name_pairs.cpp，以及一个pdf文件)

```

1 #include "name_pairs.h"
2 #include <iostream>
3
4 int main()
5 {
6     Name_pairs a, b, c;
7     a.read_names();

```

```
8     a.read_ages();
9     b.read_names();
10    b.read_ages();
11    c.read_names();
12    c.read_ages();
13
14    std::cout << "\n-----\n";
15    std::cout << "Name_pairs a:\n";
16    std::cout << a;
17    std::cout << "-----\n";
18
19    std::cout << "Name_pairs b:\n";
20    std::cout << b;
21    std::cout << "-----\n";
22
23    std::cout << "Name_pairs c:\n";
24    std::cout << c;
25    std::cout << "-----\n";
26
27    if (a == b) {
28        std::cout << "a equals to b.\n";
29    } else {
30        std::cout << "a does not equal to b.\n";
31    }
32
33    if (a != c) {
34        std::cout << "a does not equal to c.\n";
35    } else {
36        std::cout << "a equals to c.\n";
37    }
38
39    return 0;
40 }
```

提示：可以使用文件重定向避免手动输入数据。测试文本文件如下：

```
1 Alice
2 Bob
3 quit
4 20
5 30
6 Bob
7 Alice
8 quit
9 30
10 20
11 Alice
12 Bob
13 quit
14 30
15 20
```

针对上述测试数据，程序运行结果如下：

```
ryan@RyandiMac-Pro hm3 % ./another_use_name_pairs.o < name_pairs_data
Please enter a name and type "quit" to quit: Please enter a name and type "quit"
to quit: Please enter a name and type "quit" to quit: Please enter an age for A
lice: Please enter an age for Bob: Please enter a name and type "quit" to quit:
Please enter a name and type "quit" to quit: Please enter a name and type "quit"
to quit: Please enter an age for Bob: Please enter an age for Alice: Please ent
er a name and type "quit" to quit: Please enter a name and type "quit" to quit:
Please enter a name and type "quit" to quit: Please enter an age for Alice: Plea
se enter an age for Bob:
-----
Name_pairs a:
(Alice, 20)
(Bob, 30)
-----
Name_pairs b:
(Bob, 30)
(Alice, 20)
-----
Name_pairs c:
(Alice, 30)
(Bob, 20)
-----
a equals to b.
a does not equal to c.
(base) ryan@RyandiMac-Pro hm3 %
```